CSCI 447 — Machine Learning

Project #1

Assigned: August 21, 2024
Project Due: September 13, 2024

Introduction

The purpose of this assignment is to provide a gentle introduction to the field of machine learning by having you implement one algorithm and test it on a few real-world data sets. For purposes of this assignment, we won't tell you exactly what the name of the algorithm is, or provide its theoretical basis, until later. You'll just have to trust us.

Here is the algorithm you are to implement. These instructions apply to a single training data set. For this, we need some notation. We will use #{pred} to represent that we are counting the number of times the predicate given by "pred" is matched. We will give you the real notation for this algorithm in the second week of the semester.

1. For each class in the training set, calculate

$$Q(C = c_i) = \frac{\#\{\mathbf{x} \in c_i\}}{N}$$

   In other words, for each class, divide the number of examples in that class by the total number of examples N in the training set.

2. Logically, you will then separate the data into their respective classes. In reality, you don't need to do that, but it makes the explanation easier. So now think of each class having its own subset of data. We will then consider each class in turn.

3. For each attribute A_j in the class-specific training set, calculate

$$F(A_j = a_k, C = c_i) = \frac{\#\{(\mathbf{x}_{A_j} = a_k) \wedge (\mathbf{x} \in c_i)\} + 1}{N_{c_i} + d}$$

   where d is the number of attributes and $N_{c_i} = \#\{\mathbf{x} \in c_i\}$. In other words, for each attribute value, divide the number of examples that match that attribute value (plus one) by the number of examples in the class (plus d).

That's the entire training algorithm. Then to classify an example from the test set, do the following for each class. Calculate only for the attribute values a_k that exist in the example

$$C(\mathbf{x}) = Q(C = c_i) \times \prod_{j=1}^{d} F(A_j = a_k, C = c_i)$$

Then return

$$class(\mathbf{x}) = \underset{c_i \in C}{\operatorname{argmax}}\ C(\mathbf{x}).$$

In other words, return the class with the highest value for $C(\mathbf{x})$.

Data Sets

This experiment requires you to download five data sets from the UCI Machine Learning repository and train a classifier for each of these data sets. The data sets you will use are the following. They can also be found in the Assignments area under Content in Brightspace. Each data set has two files: a .NAMES file and a .DATA file. (Note that the Wine data set, should we use it, actually has two .DATA files.) The .NAMES file explains the data, providing information on all of the columns, including their types and what they represent. The .DATA file is the data itself. You do not need to be able to process the various data files in an entirely autonomous fashion, but you should try to generlize the processing as much as possible.

- Breast Cancer:

  `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\%28Original\%29`

  This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

- Glass:

  `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`

  The study of classification of types of glass was motivated by criminological investigation.

- Iris:

  `https://archive.ics.uci.edu/ml/datasets/Iris`

  The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

- Soybean (small):

  `https://archive.ics.uci.edu/ml/datasets/Soybean+\%28Small\%29`

  A small subset of the original soybean database.

- Vote:

  `https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records`

  This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac.

Issues

When using these data sets, be careful of some issues.

- Some of the data sets have missing attribute (i.e., feature) values. When this occurs in low numbers, you may simply edit the corresponding values out of the data sets. For more occurrences, you should do some kind of "data imputation" where, basically, you generate a value of some kind. This can be purely random, it can be sampled according to the conditional probability of the values occurring, given the underlying class for that example, or it can just be the mean or median value of the feature. The choice is yours, but be sure to document your choice.

- Most of the attributes in the various data sets are either multi-value discrete (categorical) or real-valued. You will need to deal with the continuous valued attributes in some way. Specifically, you will need to discretize them in some way for both algorithms (e.g., one-hot-coding after binning) and then proceed as in the multi-valued categorical case. It's up to you how you do that, but again, please document your approach.

Requirements

Design Document

You are required to submit a design document outline the design of your programming assignment. Details of what should go into your design document is the same for all of the programming assignments in this course. You can find the details of what needs to go into this document in the Content area under Administrative.

Programming

You should complete the following steps for this assignment:

- Download the five (5) data sets from the UCI Machine Learning repository. You can find this repository at `http://archive.ics.uci.edu/ml/` as well as in Brightspace. Alternatively, you may download the data sets from Brightspace.
- Pre-process the data to ensure you are working with complete examples (i.e., no missing attribute values) and discrete features only.
- Implement the above algorithm and test it on two different versions of the data. The first version is what you get from the repository without change (other than, possibly, data imputation). The second version goes through your data, selects 10% of the features at random and shuffles the values within each feature, thus introducing noise into the data.
- Select and implement at least two (2) different evaluation measures (i.e., loss functions) that you will use to evaluate your algorithm. Example loss functions include 0/1-loss, precision, recall, and F1-score.
- Develop a hypothesis for each data set based on expected performance on the different data sets.
- Design and execute experiments using 10-fold cross-validation to test your hypotheses, comparing performance on the unaltered and altered data sets from the UCI repository.

Paper

Write a very brief paper summarizing the results of your experiments. Your paper is required to be at least 5 pages and no more than 10 pages using the JMLR format. This page limit is all-inclusive of figures, tables, and references. However, the page limit does not include the Appendix listing work effort of the team. You can find templates for this format at `http://www.jmlr.org/format/format.html`. The format is also available within Overleaf. Make sure you explain the experimental setup, the tuning process, and the final parameters used for each algorithm. Your paper should contain the following elements:

- Title and author name(s)
- Problem statement, including hypothesis
- Description of your experimental approach and program design
- Presentation of the results of your experiments (in words, tables, and graphs)
- A discussion of the behavior of your algorithms, combined with any conclusions you can draw relative to your hypothesis
- Summary
- References (Only required if you use a resource other than the course content)
- Appendix listing who did what on the assignment (remember equal work balance is expected)

Video

Create a video demonstrating the functioning of your code. This video should focus on behavior and not on walking through the code. You need to show input, data structure, and output. For the video, the following constitute minimal requirements that must be satisfied:

- The video is to be no longer than 5 minutes long.
- The video should be provided in mp4 format. Alternatively, it can be uploaded to a streaming service such as YouTube with a link provided.

- Fast forwarding is permitted through long computational cycles. Fast forwarding is *not permitted* whenever there is a voice-over or when results are being presented.
- Be sure to provide verbal commentary or explanation on all of the elements you are demonstrating.
- Demonstrate your discretization method for the real-valued features.
- Show a sample trained model. This will consist of the set of class parameter values as well as the class-conditional attribute parameter values.
- Demonstrate the counting process by showing the corresponding counts for a class as well as for a class-conditional attribute counts.
- Provide sample outputs on one fold's hold-out set for one of the data sets showing classification performance on both both versions of your algorithm.
- Show the performance (based on the loss function) on one fold's hold-out set for both versions of your algorithm on one of the data sets.

Group Work

All projects are completed in teams of two or three people. Team dynamics can be difficult in that there can be an unfair balance of work completed. As an attempt to avoid this issue, the following group work requirements are put in place.

- Include a summary of the tasks completed and percent level of effort for each member of the team in accordance with the following items. This document shall be included as part of your paper submission. The level of effort will be used to determine weighting of the project grade. Specifically, if a team member does less than 40% (for a two-person team) or 25% (for a three-person team), their grade will be adjusted downward to reflect their level of effort.
- There are three major components to each project—code, paper, and video. A division of labor where one person is responsible for code, one person for the paper, and one person for the video is *not allowed*. Everyone on the team is required to contribute to all three of the components.
- The coding can be subdivided based on the individual coding requirements (e.g., data preparation, algorithms, experimental framework, analysis code etc.). In this assignment, there is only one algorithm to implement, so care should be given to make sure the coding requirements are distributed equitably.
- The paper should be subdivided based on paper section. It is not sufficient for a member of the team to write the entire paper or just to proofread the paper. The goal is for each member of the team to contribute equal content to the final paper.
- The video should be recorded by a single person; however, all must contribute to the development of the video (e.g., script, setup for a particular item to be shown, editing of the final video). For the video, a different person should do the recording from project to project.

Submission

Submit your fully documented code with the outputs from running your programs, your video, and your paper. Your submission should include the following components and should be submitted as a "group" submission (i.e., only one submission for the entire group).

- This assignment requires you to submit a design document two weeks before submitting the program, video, and paper. Details on what needs to go into the design document are provided in the Content Area. Your design docment will be checked using the TurnItIn plagiarism checking system,. Submit to the "P1 Design Document" portion of the assignment. This document can be formatted however you wish but must be submitted as a PDF file. Your design document is worth 30% of the grade.

- Zip your source code files into a single zip file. Do not use alternative or archive formats like tar, gz, rar, etc. Be sure to include the team member contribution report with your code. Submit to the "P1 Code" portion of the assignment. The code must be well structured and fully commented. Code is worth 10% of the overall project grade.
- Submit a PDF of your report to the "P1 Paper" portion of the assignment. Brightspace is set up to accept PDF files only. Your report will be checked using the TurnItIn plagiarism checking system. The paper must be formatted per JMLR formatting requirements. All math should be included using the math editing capabilities of your document processing tool. Figures needs to be high quality. Do not use screenshots. Your paper is worth 40% of the grade.
- Submit either your video or a text file containing a link to your video in the "P1 Video" portion of the assignment. Note that a file is required to be submitted, so if you are using a streaming service, remember to submit that text file. Your video is limited to five minutes maximum. Anything that appears in the video beyond five minutes will be ignored. Use the video the satisfy the requirements above, rather than "walking through" the code. The video is worth 20% of the grade.